

**TRANSMITTAL OF APPEAL BRIEF (Large Entity)**Docket No.  
200-0664In Re Application Of: **Joseph G. Walacavage et al.**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
09/965,905	September 28, 2001	J. Proctor	33481	2123	4248

Invention: **METHOD OF PART FLOW MODEL FOR  
PROGRAMMABLE LOGIC CONTROLLER LOGICAL  
VERIFICATION SYSTEM****COMMISSIONER FOR PATENTS:**Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed on:  
**March 31, 2008**The fee for filing this Appeal Brief is: **\$510.00**

- ☐ A check in the amount of the fee is enclosed.
- ☐ The Director has already been authorized to charge fees in this application to a Deposit Account.
- ☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 06-1510. I have enclosed a duplicate copy of this sheet.
- ☐ Payment by credit card. Form PTO-2038 is attached.

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**  
SignatureDated: **June 2, 2008****Daniel H. Bliss (Reg. No. 32,398) [0693.00242]  
Bliss McGlynn, P.C.  
2075 West Big Beaver Road, Suite 600  
Troy, Michigan 48084  
(248) 649-6090****Record I.D. 81082348**

cc:

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on

June 2, 2008

(Date)

  
Signature of Person Mailing Correspondence**Daniel H. Bliss**

Typed or Printed Name of Person Mailing Correspondence



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Art Unit: 2123 )  
Examiner: J. Proctor )  
Applicant(s): J. G. Walacavage et al. )  
Serial No.: 09/965,905 )  
Filing Date: September 28, 2001 )  
For: METHOD OF PART FLOW MODEL FOR )  
PROGRAMMABLE LOGIC CONTROLLER )  
LOGICAL VERIFICATION SYSTEM )

**APPEAL BRIEF**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, Virginia 22313-1450

Sir:

By Notice of Appeal filed March 31, 2008, Applicants have appealed the Final Rejection dated October 30, 2007 and submit this brief in support of that appeal.

**REAL PARTY IN INTEREST**

The real party in interest is the Assignee, Ford Global Technologies, Inc.

**RELATED APPEALS AND INTERFERENCES**

There are no related appeals or interferences regarding the present application.

CERTIFICATE OF MAILING: (37 C.F.R. 1.8) I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service with sufficient postage as First Class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450 on June 2, 2008, by Daniel H. Bliss.

### **STATUS OF CLAIMS**

Claims 1 through 8 have been rejected.

Claim 9 has been canceled.

Claim 10 has been rejected.

Claim 11 has been canceled.

Claims 12 through 21 have been rejected.

Claims 1 through 8, 10, and 12 through 21 are being appealed.

### **STATUS OF AMENDMENTS**

An Amendment Under 37 C.F.R. 1.116 was filed on January 30, 2008 in response to the Final Office Action dated October 30, 2007. An Advisory Action dated February 20, 2008 was issued and indicated that the Amendment under 37 C.F.R. 1.116 did not place the application in a condition for allowance. The Advisory Action stated that the amendments raised new issues that would require further consideration and/or search and did not indicate whether the Amendment under 37 C.F.R. 1.116 would be entered upon filing an appeal. A Notice of Appeal, along with the requisite fee, was filed on March 31, 2008. The Appeal Brief, along with the requisite fee, is submitted herewith.

### **SUMMARY OF THE CLAIMED SUBJECT MATTER**

#### **Independent Claim 1**

The claimed subject matter of independent claim 1 is directed to a method of part flow for a programmable logic controller logical verification system. [Referring to FIG. 2, a

method, according to the present invention, for application of part flow model as part of the PLC logical verification system 18 is shown. In general, the user 12 identifies part locations, including movement between stationary locations, on a VPLC workspace of the computer 14 using a part location editor of the computer 14. Each location has the capability of having resources attached to it, including part location switches. The collection of part locations make up a directed graph that, coupled with a part generator, allows the user 12 to visually see the flow of parts through the PLC logical verification system 18 by change of color (indicating the presence of a part) at any of the part locations. It should be appreciated that, once a basic part flow model has been implemented, the method may be extended in more elaborate data movement schemes.] (FIGS. 1 and 2; Specification, page 8, line 1 through 16).

The method includes the steps of constructing a simulation model of a manufacturing line using a computer. [The method includes writing a control model file for part flow by the part flow design system 20. For example, the part flow design system 20 will create a part flow model definition that describes how a part flows through a workcell such as moving from a bin into a fixture. It should be appreciated that the part flow model is information that describes events, dependencies, and logical conditions that represent or simulate part flow through the workcell. The method writes a control model file by the PLC logical verification system 18 to “logically link” the fixtures of the fixture design system 16a and the workcells of the workcell design system 16b and the part flow of the part flow design system 20 into a tooling or manufacturing line.] (FIGS. 1 and 2; Specification, page 9, line 16 through page 10, line 2 and page 10, lines 10 through 14).

The method also includes the steps of playing the simulation model by a PLC logical verification system on the computer and viewing a flow of a part through the

manufacturing line by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. [After the part flow model is designed, the method includes playing the part flow model by the PLC logical verification system 18. For example, the user 12 plays the part flow model by the PLC logical verification system 18 on the computer 14. The collection of part locations make up a directed graph that, coupled with a part generator, allows the user 12 to visually see the flow of parts through the PLC logical verification system 18 by change of color (indicating the presence of a part) at any of the part locations. The user 12 tests the logic by forcing a state in the control logic to test all exception logic. For example, the method tests for status as to whether the part is present or not present. It should also be appreciated that a record exists with each part generated and that the individual resources can contribute information to the part record (such as an action performed or another part being bound to it). It should further be appreciated that the unique part record can be tested as it traverses the workcell, which allows subsystem capabilities such as quality and routing to be exercised. It should still further be appreciated that the method is an iterative process between design and simulation carried out on the computer 14 by the user 12. The PLC logical verification system 18 verifies the PLC logic for a workcell of a tooling or manufacturing line. The computer 14 also sends and receives information with a part flow design 20 via an electronic link. The part flow design 20 sends and receives information with the PLC logical verification system 18 to verify the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 3 through 7 and page 8, lines 8 through 13 and page 12, line 8, through page 13, line 2 and page 7, lines 8 through 13).

The method further includes the steps of determining if the part flow represented in the simulation model is correct to the user. [The method includes determining whether the

part flow model is acceptable. For example, the user 12 determines whether the part has traversed the workcell successfully. If the part flow model is not acceptable, the method includes modifying the part flow model. The user 12 uses the iterative process to change resources and capabilities of the part record and runs or simulates the part flow model with the PLC logical verification system 18 until it is acceptable to the user 12.] (FIGS. 1 and 2; Specification, page 13, lines 7 through 13).

The method still further includes the steps of generating the PLC code if the part flow represented in the simulation model is correct. [Once the part flow model is acceptable to the user 12, the method includes generating PLC code and using the PLC code to build a manufacturing line. It should be appreciated that the part flow model is similar to a floor plan and is the basis for the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 15 through 19).

The method still further includes the steps of using the generated PLC code and implementing the manufacturing line according to the part flow simulation model. [Once the part flow model is acceptable to the user 12, the method includes generating PLC code and using the PLC code to build a manufacturing line. It should be appreciated that the part flow model is similar to a floor plan and is the basis for the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 15 through 19).

#### **Independent claim 12**

The claimed subject matter of independent claim 12 is directed to a method for application of a part flow for a programmable logic controller logical verification system. [Referring to FIG. 2, a method, according to the present invention, for application of part flow model as part of the PLC logical verification system 18 is shown. In general, the user 12

identifies part locations, including movement between stationary locations, on a VPLC workspace of the computer 14 using a part location editor of the computer 14. Each location has the capability of having resources attached to it, including part location switches. The collection of part locations make up a directed graph that, coupled with a part generator, allows the user 12 to visually see the flow of parts through the PLC logical verification system 18 by change of color (indicating the presence of a part) at any of the part locations. It should be appreciated that, once a basic part flow model has been implemented, the method may be extended in more elaborate data movement schemes.] (FIGS. 1 and 2; Specification, page 8, line 1 through 16).

The method includes the steps of constructing a simulation model of a part flow in a manufacturing line using a computer by representing a part and part locations of the manufacturing line. [The method includes writing a control model file for part flow by the part flow design system 20. For example, the part flow design system 20 will create a part flow model definition that describes how a part flows through a workcell such as moving from a bin into a fixture. It should be appreciated that the part flow model is information that describes events, dependencies, and logical conditions that represent or simulate part flow through the workcell. The method writes a control model file by the PLC logical verification system 18 to “logically link” the fixtures of the fixture design system 16a and the workcells of the workcell design system 16b and the part flow of the part flow design system 20 into a tooling or manufacturing line.] (FIGS. 1 and 2; Specification, page 9, line 16 through page 10, line 2 and page 10, lines 10 through 14).

The method also includes the steps of playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line and viewing a flow of the represented part through the

manufacturing line by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. [After the part flow model is designed, the method includes playing the part flow model by the PLC logical verification system 18. For example, the user 12 plays the part flow model by the PLC logical verification system 18 on the computer 14. The collection of part locations make up a directed graph that, coupled with a part generator, allows the user 12 to visually see the flow of parts through the PLC logical verification system 18 by change of color (indicating the presence of a part) at any of the part locations. The user 12 tests the logic by forcing a state in the control logic to test all exception logic. For example, the method tests for status as to whether the part is present or not present. It should also be appreciated that a record exists with each part generated and that the individual resources can contribute information to the part record (such as an action performed or another part being bound to it). It should further be appreciated that the unique part record can be tested as it traverses the workcell, which allows subsystem capabilities such as quality and routing to be exercised. It should still further be appreciated that the method is an iterative process between design and simulation carried out on the computer 14 by the user 12. The PLC logical verification system 18 verifies the PLC logic for a workcell of a tooling or manufacturing line. The computer 14 also sends and receives information with a part flow design 20 via an electronic link. The part flow design 20 sends and receives information with the PLC logical verification system 18 to verify the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 3 through 7 and page 8, lines 8 through 13 and page 12, line 8, through page 13, line 2 and page 7, lines 8 through 13).

The method includes the steps of determining if the part flow represented in the simulation model is correct to the user. [The method includes determining whether the part flow



model is acceptable. For example, the user 12 determines whether the part has traversed the workcell successfully. If the part flow model is not acceptable, the method includes modifying the part flow model. The user 12 uses the iterative process to change resources and capabilities of the part record and runs or simulates the part flow model with the PLC logical verification system 18 until it is acceptable to the user 12.] (FIGS. 1 and 2; Specification, page 13, lines 7 through 13).

The method further includes the steps of generating the PLC code if the part flow simulation model is correct. [Once the part flow model is acceptable to the user 12, the method includes generating PLC code and using the PLC code to build a manufacturing line. It should be appreciated that the part flow model is similar to a floor plan and is the basis for the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 15 through 19).

The method further includes the steps of using the generated PLC code and implementing the manufacturing line according to the part flow simulation model. [Once the part flow model is acceptable to the user 12, the method includes generating PLC code and using the PLC code to build a manufacturing line. It should be appreciated that the part flow model is similar to a floor plan and is the basis for the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 15 through 19).

#### **Independent claim 21**

The claimed subject matter of independent claim 21 is directed to a method for application of a part flow for a programmable logic controller logical verification system. [Referring to FIG. 2, a method, according to the present invention, for application of part flow model as part of the PLC logical verification system 18 is shown. In general, the user 12

identifies part locations, including movement between stationary locations, on a VPLC workspace of the computer 14 using a part location editor of the computer 14. Each location has the capability of having resources attached to it, including part location switches. The collection of part locations make up a directed graph that, coupled with a part generator, allows the user 12 to visually see the flow of parts through the PLC logical verification system 18 by change of color (indicating the presence of a part) at any of the part locations. It should be appreciated that, once a basic part flow model has been implemented, the method may be extended in more elaborate data movement schemes.] (FIGS. 1 and 2; Specification, page 8, line 1 through 16).

The method includes the steps of constructing a simulation model of a part flow in a manufacturing line using a computer by selecting a part generator, generating a part with the part generator, and identifying part locations of the part in the manufacturing line. [The method includes writing a control model file for part flow by the part flow design system 20. For example, the part flow design system 20 will create a part flow model definition that describes how a part flows through a workcell such as moving from a bin into a fixture. It should be appreciated that the part flow model is information that describes events, dependencies, and logical conditions that represent or simulate part flow through the workcell. The method writes a control model file by the PLC logical verification system 18 to “logically link” the fixtures of the fixture design system 16a and the workcells of the workcell design system 16b and the part flow of the part flow design system 20 into a tooling or manufacturing line. The method starts with a part generator in bubble 100. The part generator is a representation of some part such as a vehicle quarter panel selected by the user 12. The part generator generates a part type such as a front driver side quarter panel and serial number such as body style. From bubble 100, the method advances to block 102 and the part generator generates a unique part from a part location

such as a bin of the parts. The method advances to bubble 104 and moves the part from the bin to another part location. ] (FIGS. 1 and 2; Specification, page 9, line 16 through page 10, line 2 and page 10, lines 10 through 14 and page 10, line 20 through page 11, line 7).

The method also includes the steps of playing the simulation model of the part flow by a PLC logical verification system on the computer to move the generated part to and from locations within the manufacturing line and viewing a flow of the part through the manufacturing line by a change of color at any of the part locations by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. [After the part flow model is designed, the method includes playing the part flow model by the PLC logical verification system 18. For example, the user 12 plays the part flow model by the PLC logical verification system 18 on the computer 14. The collection of part locations make up a directed graph that, coupled with a part generator, allows the user 12 to visually see the flow of parts through the PLC logical verification system 18 by change of color (indicating the presence of a part) at any of the part locations. The user 12 tests the logic by forcing a state in the control logic to test all exception logic. For example, the method tests for status as to whether the part is present or not present. It should also be appreciated that a record exists with each part generated and that the individual resources can contribute information to the part record (such as an action performed or another part being bound to it). It should further be appreciated that the unique part record can be tested as it traverses the workcell, which allows subsystem capabilities such as quality and routing to be exercised. It should still further be appreciated that the method is an iterative process between design and simulation carried out on the computer 14 by the user 12. The PLC logical verification system 18 verifies the PLC logic for a workcell of a tooling or manufacturing line.

The computer 14 also sends and receives information with a part flow design 20 via an electronic link. The part flow design 20 sends and receives information with the PLC logical verification system 18 to verify the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 3 through 7 and page 8, lines 8 through 13 and page 12, line 8, through page 13, line 2 and page 7, lines 8 through 13).

The method includes the steps of determining if the part flow represented in the simulation model is correct to the user. [The method includes determining whether the part flow model is acceptable. For example, the user 12 determines whether the part has traversed the workcell successfully.] (FIGS. 1 and 2; Specification, page 13, lines 7 through 10).

The method includes the steps of modifying the part flow represented in the simulation model if the part flow represented in the simulation model is not correct. [If the part flow model is not acceptable, the method includes modifying the part flow model. The user 12 uses the iterative process to change resources and capabilities of the part record and runs or simulates the part flow model with the PLC logical verification system 18 until it is acceptable to the user 12.] (FIGS. 1 and 2; Specification, page 13, lines 10 through 15).

The method includes the steps of generating the PLC code if the part flow simulation model is correct. [Once the part flow model is acceptable to the user 12, the method includes generating PLC code and using the PLC code to build a manufacturing line. It should be appreciated that the part flow model is similar to a floor plan and is the basis for the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 15 through 19).

The method further includes the steps of using the generated PLC code and implementing the manufacturing line according to the part flow simulation model. [Once the part flow model is acceptable to the user 12, the method includes generating PLC code and using the

PLC code to build a manufacturing line. It should be appreciated that the part flow model is similar to a floor plan and is the basis for the PLC code.] (FIGS. 1 and 2; Specification, page 13, lines 15 through 19).

### **GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

The first ground of rejection to be reviewed on appeal is whether the claimed invention of claims 1, 10, 12, 20, and 21 is disclosed and anticipated under 35 U.S.C. § 102(b) by “Simulation, Animation, and Shop-Floor Control”, by Cynthia Erickson et al.

The second ground of rejection to be reviewed on appeal is whether the claimed invention of claims 2 through 8 and 13 through 19 is obvious and unpatentable under 35 U.S.C. § 103 over Erickson et al. in view of “Emulation of a Material Delivery System”, by Todd LeBaron and Kelly Thompson.

### **ARGUMENT**

#### **Claims Not Disclosed or Anticipated Under 35 U.S.C. § 102**

As to patentability, 35 U.S.C. § 102(b) provides that a person shall be entitled to a patent unless:

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

A rejection grounded on anticipation under 35 U.S.C. § 102 is proper only where the subject matter claimed is identically disclosed or described in a reference. In other words, anticipation requires the presence of a single prior art reference which discloses each and every

element of the claimed invention arranged as in the claim. In re Arkley, 455 F.2d 586, 172 U.S.P.Q. 524 (C.C.P.A. 1972); Kalman v. Kimberly-Clark Corp., 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983); Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick Co., 730 F.2d 1452, 221 U.S.P.Q. 481 (Fed. Cir. 1984).

As to the reference applied by the Examiner under 35 U.S.C. § 102(b), the publication “Simulation, Animation, and Shop-Floor Control”, by Cynthia Erickson et al. discloses that it is often desirable to link shop-floor hardware directly to a discrete-event simulation model or graphical animation. In this case, a simulation model, linked directly to one or more programmable controllers, provides the controller(s) with system scenarios under which they are expected to perform and produce a response. A second application is system emulation, where real-time data is used to drive an animation. Emulation has been used to describe graphical systems displaying the current status of the manufacturing shop floor. Linking a simulation directly to a programmable logic controller (PLC) provides a means to test the control logic of the PLC. Once the control logic for the PLC has been written, it must be debugged and tested. Currently, much of this verification takes place on the shop floor once the manufacturing system is in place. A startup phase is usually planned, during which the manufacturing system operates at low capacity, in order to debug the control software. To verify PLC logic using simulation, a model of the physical system must be developed; however, the timing of some events would be generated by the PLC.

#### **Claims 1 and 10**

Independent claim 1 claims the present invention claimed as a method of part flow for a programmable logic controller logical verification system. The method includes the steps of

constructing a simulation model of a manufacturing line using a computer, playing the simulation model by a PLC logical verification system on the computer, and viewing a flow of a part through the manufacturing line by a user. The PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. The method also includes the steps of determining if the part flow represented in the simulation model is correct to the user and generating PLC code if the part flow represented in the simulation model is correct. The method further includes the steps of using the generated PLC code and implementing the manufacturing line according to the part flow simulation model.

As to claim 10, claim 10 claims the method as set forth in claim 1 including the step of modifying the part flow represented in the simulation model if the part flow represented in the simulation model is not correct.

Erickson et al. does not disclose or anticipate the claimed invention of claims 1 and 10. Specifically, Erickson et al. merely discloses linking shop-floor hardware such as programmable controllers directly to a discrete-event simulation model or a graphical animation using a system emulator. Erickson et al. lacks playing a simulation model by a PLC logical verification system on a computer and viewing a flow of a part through a manufacturing line by a user, wherein the PLC logical verification system dynamically interacting through input and output with the simulation model to verify a PLC code of the manufacturing line. In Erickson et al., there is no PLC logical verification system and the simulation model is linked directly with one or more hardware PLCs to test the control logic of the PLC. Also in Erickson et al., one or more hardware PLCs are linked with a system emulator for graphical animation of the manufacturing shop floor. As is known in the art, an emulator represents a physical device in

software. (See Wikipedia dictionary definition from Wikipedia website). The PLC logical verification system is not a software representation of a physical device, but a software tool that allows dynamic interaction directly with a simulation model to test PLC logic by having an input and output exchange similar to input/output control logic to validate that the logic is delivering what is intended. The PLC logical verification system analytically verifies the PLC logic. (A similar system is disclosed in U.S. Patent No. 6,442,441). In Erickson et al., the emulator does not allow dynamic interaction directly with a simulation model to test PLC logic to verify a PLC code of the manufacturing line.

Based on the above, Erickson et al. fails to disclose the combination of a method of part flow for a programmable logic controller logical verification system including the steps of constructing a simulation model of a part flow in a manufacturing line using a computer, playing the simulation model by a PLC logical verification system on the computer, viewing a flow of a part through the manufacturing line by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line, determining if the part flow represented in the simulation model is correct to the user, generating PLC code if the part flow represented in the simulation model is correct, using the generated PLC code, and implementing the manufacturing line according to the part flow simulation model as claimed by Applicants in independent claim 1 and dependent claim 10. Erickson et al. fails to disclose each and every element of the claimed combination of a method of part flow for a programmable logic controller logical verification system as arranged in the claims and claimed by Applicants. As a result, the Erickson et al. publication cannot be an anticipatory reference under 35 U.S.C. § 102(b) to claims 1 and 10 of the present application.



Against this background, it is submitted that the present invention is not anticipated in view of the disclosure of Erickson et al. The reference fails to disclose each and every element of the claimed combination of a method of part flow for a programmable logic controller logical verification system including a PLC logical verification system dynamically interacting through input and output with a simulation model to verify a PLC code of a manufacturing line as claimed by Applicants. Therefore, it is respectfully submitted that claims 1 and 10 are not anticipated and are allowable over the rejection under 35 U.S.C. § 102(b).

#### **Claims 12 and 20**

Independent claim 12 claims the present invention as a method for application of a part flow for a programmable logic controller logical verification system. The method includes the steps of constructing a simulation model of a part flow in a manufacturing line using a computer by representing a part and part locations of the manufacturing line. The method also includes the steps of playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line, and viewing a flow of the represented part through the manufacturing line by a user , wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. The method further includes the steps of determining if the part flow represented in the simulation model is correct to the user, generating the PLC code if the part flow simulation model is correct, and using the generated PLC code and implementing the manufacturing line according to the part flow simulation model.

As to claim 20, claim 20 claims a method as set forth in claim 1 {Sic 12} including the step of modifying the part flow represented in the simulation model if the part flow represented in the simulation model is not correct. [Note: Claim 20 is recited as being dependent on claim 1, but should be dependent on claim 12 to avoid duplication and Counsel for Applicants respectfully requests the Examiner to change the dependency by Examiner's Amendment.]

Erickson et al. does not disclose or anticipate the claimed invention of claims 12 and 20. Specifically, Erickson et al. merely discloses linking shop-floor hardware such as programmable controllers directly to a discrete-event simulation model or a graphical animation using a system emulator. Erickson et al. lacks playing a simulation model by a PLC logical verification system on a computer to move a represented part to and from part locations within a manufacturing line and viewing a flow of the represented part through the manufacturing line by a user, wherein the PLC logical verification system dynamically interacting through input and output with the simulation model to verify a PLC code of the manufacturing line. In Erickson et al., there is no PLC logical verification system and the simulation model is linked directly with one or more hardware PLCs to test the control logic of the PLC. Also in Erickson et al., one or more hardware PLCs are linked with a system emulator for graphical animation of the manufacturing shop floor. As is known in the art, an emulator represents a physical device in software. (See Wikipedia dictionary definition from Wikipedia website). The PLC logical verification system is not a software representation of a physical device, but a software tool that allows dynamic interaction directly with a simulation model to test PLC logic by having an input and output exchange similar to input/output control logic to validate that the logic is delivering what is intended. The PLC logical verification system analytically verifies the PLC logic. (A similar system is disclosed in U.S. Patent No. 6,442,441). In Erickson et al., the emulator does

not allow dynamic interaction directly with a simulation model to test PLC logic to verify a PLC code of the manufacturing line.

Based on the above, Erickson et al. fails to disclose the combination of a method of part flow for a programmable logic controller logical verification system including the steps of constructing a simulation model of a part flow in a manufacturing line using a computer by representing a part and part locations of the manufacturing line, playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line, viewing a flow of the represented part through the manufacturing line by a user , wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line, determining if the part flow represented in the simulation model is correct to the user, generating the PLC code if the part flow simulation model is correct, and using the generated PLC code and implementing the manufacturing line according to the part flow simulation model as claimed by Applicants in independent claim 12 and dependent claim 20. Erickson et al. fails to disclose each and every element of the claimed combination of a method of part flow for a programmable logic controller logical verification system as arranged in the claims and claimed by Applicants. As a result, the Erickson et al. publication cannot be an anticipatory reference under 35 U.S.C. § 102(b) to claims 12 and 20 of the present application.

Against this background, it is submitted that the present invention is not anticipated in view of the disclosure of Erickson et al. The reference fails to disclose each and every element of the claimed combination of a method of part flow for a programmable logic controller logical verification system including playing a simulation model by a PLC logical verification system on a computer to move a represented part to and from part locations within a

manufacturing line and viewing a flow of the represented part through the manufacturing line by a user , wherein the PLC logical verification system dynamically interacting through input and output with the simulation model to verify a PLC code of the manufacturing line as claimed by Applicants. Therefore, it is respectfully submitted that claims 12 and 20 are not anticipated and are allowable over the rejection under 35 U.S.C. § 102(b).

### **Claim 21**

Independent claim 21 claims a method for application of a part flow for a programmable logic controller logical verification system. The method includes the steps of constructing a simulation model of a part flow in a manufacturing line using a computer by selecting a part generator, generating a part with the part generator, and identifying part locations of the part in the manufacturing line. The method also includes the steps of playing the simulation model of the part flow by a PLC logical verification system on the computer to move the generated part to and from locations within the manufacturing line and viewing a flow of the part through the manufacturing line by a change of color at any of the part locations by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. The method includes the steps of determining if the part flow represented in the simulation model is correct to the user and modifying the part flow represented in the simulation model if the part flow represented in the simulation model is not correct. The method further includes the steps of generating the PLC code if the part flow simulation model is correct and using the generated PLC code and implementing the manufacturing line according to the part flow simulation model.

Erickson et al. does not disclose or anticipate the claimed invention of claim 21. Specifically, Erickson et al. merely discloses linking shop-floor hardware such as programmable controllers directly to a discrete-event simulation model or a graphical animation using a system emulator. Erickson et al. lacks constructing a simulation model of a part flow in a manufacturing line using a computer by selecting a part generator, generating a part with the part generator, and identifying part locations of the part in the manufacturing line. Erickson et al. also lacks playing the simulation model of the part flow by a PLC logical verification system on the computer to move the generated part to and from locations within the manufacturing line and viewing a flow of the part through the manufacturing line by a change of color at any of the part locations by a user, wherein the PLC logical verification system dynamically interacting through input and output with the simulation model to verify a PLC code of the manufacturing line. In Erickson et al., there is no PLC logical verification system and the simulation model is linked directly with one or more hardware PLCs to test the control logic of the PLC. Also in Erickson et al., one or more hardware PLCs are linked with a system emulator for graphical animation of the manufacturing shop floor. As is known in the art, an emulator represents a physical device in software. (See Wikipedia dictionary definition from Wikipedia website). The PLC logical verification system is not a software representation of a physical device, but a software tool that allows dynamic interaction directly with a simulation model to test PLC logic by having an input and output exchange similar to input/output control logic to validate that the logic is delivering what is intended. The PLC logical verification system analytically verifies the PLC logic. (A similar system is disclosed in U.S. Patent No. 6,442,441). In Erickson et al., the emulator does not allow dynamic interaction directly with a simulation model to test PLC logic to verify a PLC code of the manufacturing line.

Based on the above, Erickson et al. fails to disclose the combination of a method of part flow for a programmable logic controller logical verification system including the steps of constructing a simulation model of a part flow in a manufacturing line using a computer by selecting a part generator, generating a part with the part generator, identifying part locations of the part in the manufacturing line, playing the simulation model of the part flow by a PLC logical verification system on the computer to move the generated part to and from locations within the manufacturing line, viewing a flow of the part through the manufacturing line by a change of color at any of the part locations by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line, determining if the part flow represented in the simulation model is correct to the user, modifying the part flow represented in the simulation model if the part flow represented in the simulation model is not correct, generating the PLC code if the part flow simulation model is correct, and using the generated PLC code and implementing the manufacturing line according to the part flow simulation model as claimed by Applicants in independent claim 21. Erickson et al. fails to disclose each and every element of the claimed combination of a method of part flow for a programmable logic controller logical verification system as arranged in the claims and claimed by Applicants. As a result, the Erickson et al. publication cannot be an anticipatory reference under 35 U.S.C. § 102(b) to claim 21 of the present application.

Against this background, it is submitted that the present invention is not anticipated in view of the disclosure of Erickson et al. The reference fails to disclose each and every element of the claimed combination of a method of part flow for a programmable logic controller logical verification system constructing a simulation model of a part flow in a

manufacturing line using a computer by selecting a part generator, generating a part with the part generator, identifying part locations of the part in the manufacturing line, playing the simulation model of the part flow by a PLC logical verification system on the computer to move the generated part to and from locations within the manufacturing line, viewing a flow of the part through the manufacturing line by a change of color at any of the part locations by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line as claimed by Applicants. Therefore, it is respectfully submitted that claim 21 is not anticipated and are allowable over the rejection under 35 U.S.C. § 102(b).

**Claims Not Obvious or Unpatentable Under 35 U.S.C. § 103**

As to patentability, 35 U.S.C. § 103 provides that a patent may not be obtained:

If the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Id.

The United States Supreme Court interpreted the standard for 35 U.S.C. § 103 in Graham v. John Deere, 383 U.S. 1, 148 U.S.P.Q. 459 (1966). In Graham, the Court stated that under 35 U.S.C. § 103:

The scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or non-obviousness of the subject matter is determined. 148 U.S.P.Q. at 467.

Using the standard set forth in Graham, the scope and content of the prior art relied upon by the Examiner will be determined.

As to the scope and content of the prior art applied by the Examiner, the publication “Simulation, Animation, and Shop-Floor Control”, by Cynthia Erickson et al. discloses that it is often desirable to link shop-floor hardware directly to a discrete-event simulation model or graphical animation. In this case, a simulation model, linked directly to one or more programmable controllers, provides the controller(s) with system scenarios under which they are expected to perform and produce a response. A second application is system emulation, where real-time data is used to drive an animation. Emulation has been used to describe graphical systems displaying the current status of the manufacturing shop floor. Linking a simulation directly to a programmable logic controller (PLC) provides a means to test the control logic of the PLC. Once the control logic for the PLC has been written, it must be debugged and tested. Currently, much of this verification takes place on the shop floor once the manufacturing system is in place. A startup phase is usually planned, during which the manufacturing system operates at low capacity, in order to debug the control software. To verify PLC logic using simulation, a model of the physical system must be developed; however, the timing of some events would be generated by the PLC.

The publication, “Emulation of a Material Delivery System”, by Todd LeBaron and Kelly Thompson, discloses emulation of a complex pick and pack system. A material handling system consists of conveyor sections which continuously move carriers around a closed loop that connects all pick and pack stations. Routing logic, PLC or PC control software, sequencing algorithms and more can be integrated, tested, and debugged within a simulation environment. Emulation has been used for a Rapistan Systems Project to test, debug, and



optimize complex algorithms and control logic. Emulation of the Rapistan control system for this project integrates a simulation model with the actual control system. The simulation model provides the output for evaluating control logic and algorithms. The emulation used at Rapistan Systems was able to prove that the system could handle the projected growth in daily orders. Emulation provides the graphical and statistical output needed to accurately evaluate different algorithms and control logic.

### **Claims 2 through 8**

In contradistinction, claim 1 claims the present invention claimed as a method of part flow for a programmable logic controller logical verification system. The method includes the steps of constructing a simulation model of a manufacturing line using a computer, playing the simulation model by a PLC logical verification system on the computer, and viewing a flow of a part through the manufacturing line by a user. The PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. The method also includes the steps of determining if the part flow represented in the simulation model is correct to the user and generating PLC code if the part flow represented in the simulation model is correct. The method further includes the steps of using the generated PLC code and implementing the manufacturing line according to the part flow simulation model.

Claim 2 claims the method as set forth in claim 1 wherein the step of constructing comprises selecting a part generator. Claim 3 claims the method as set forth in claim 2 wherein the step of constructing further comprises generating the part with the part generator. Claim 4 claims the method as set forth in claim 3 wherein the step of constructing further comprises

identifying part locations of the generated part within the manufacturing line. Claim 5 claims the method as set forth in claim 4 wherein the step of constructing further comprises testing the generated part at the part locations. Claim 6 claims the method as set forth in claim 1 wherein the step of constructing comprises constructing a record for the part. Claim 7 claims the method as set forth in claim 6 wherein the record has at least one resource. Claim 8 claims the method as set forth in claim 7 wherein the at least one resource has at least one capability.

As to the differences between the prior art and the claims at issue, Erickson et al. merely discloses linking shop-floor hardware such as programmable controllers directly to a discrete-event simulation model or a graphical animation using a system emulator. Erickson et al. lacks selecting a part generator, generating the part with the part generator, identifying part locations of the generated part within the manufacturing line, testing the generated part at the part locations, constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability. Erickson et al. also lacks playing a simulation model by a PLC logical verification system on a computer and viewing a flow of a part through a manufacturing line by a user, wherein a PLC logical verification system dynamically interacting through input and output with the simulation model to verify a PLC code of the manufacturing line. In Erickson et al., there is no PLC logical verification system and the simulation model is linked directly with one or more hardware PLCs to test the control logic of the PLC. Also in Erickson et al., one or more hardware PLCs are linked with a system emulator for graphical animation of the manufacturing shop floor. As is known in the art, an emulator represents a physical device in software. (See Wikipedia dictionary definition from Wikipedia website). The PLC logical verification system is not a software representation of a physical device, but a software tool that allows dynamic interaction directly with a simulation model to

test PLC logic by having an input and output exchange similar to input/output control logic to validate that the logic is delivering what is intended. The PLC logical verification system analytically verifies the PLC logic. (A similar system is disclosed in U.S. Patent No. 6,442,441).

In Erickson et al., the emulator does not allow dynamic interaction directly with a simulation model to test PLC logic to verify a PLC code of the manufacturing line.

LeBaron et al. merely discloses an emulation of a material delivery system in which routing logic, PLC or PC control software, sequencing algorithms and more can be integrated, tested, and debugged within a simulation environment. LeBaron et al. lacks selecting a part generator, generating the part with the part generator, identifying part locations of the generated part within the manufacturing line, testing the generated part at the part locations, constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability. LeBaron et al. also lacks playing a simulation model by a PLC logical verification system on a computer and viewing a flow of a part through the manufacturing line by the user wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. In LeBaron et al., there is no PLC logical verification system and no PLC code is generated. As is known in the art, an emulator represents a physical device in software. (See Wikipedia dictionary definition from Wikipedia website). The PLC logical verification system is not a software representation of a physical device, but a software tool that allows dynamic interaction directly with a simulation model to test PLC logic by having an input and output exchange similar to input/output control logic to validate that the logic is delivering what is intended. The PLC logical verification system analytically verifies the PLC logic. (A similar

system is disclosed in U.S. Patent No. 6,442,441). In LeBaron et al., the emulator does not verify the PLC logic or validate that the logic is delivering what is intended.

LeBaron et al. further lacks using the generated PLC code and implementing the manufacturing line according to the part flow simulation model. In LeBaron et al., while LeBaron et al. mentions PLC or PC control software can be tested and debugged within a simulation environment, there is no part flow for a programmable logic controller logical verification system that is viewed by a user and determining if the part flow represented in the simulation model is correct to the user. Further, in LeBaron et al., there is no generated PLC code that is used in implementing a manufacturing line.

As to a level of ordinary skill in the art, Erickson et al. discloses linking shop-floor hardware such as programmable controllers directly to a discrete-event simulation model or a graphical animation using a system emulator. LeBaron et al. discloses emulation in a delivery system in which routing logic, PLC or PC control software, sequencing algorithms and more can be integrated, tested, and debugged within a simulation environment. There is absolutely no teaching of a level of skill in the programmable logic controller and vehicle manufacturing art that a method of part flow for a programmable logic controller logical verification system includes selecting a part generator, generating the part with the part generator, identifying part locations of the generated part within the manufacturing line, testing the generated part at the part locations, constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability, playing a simulation model by a PLC logical verification system on a computer and viewing a flow of a part through the manufacturing line by a user wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. The

Examiner may not, because he doubts that the invention is patentable, resort to speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis. See In re Warner, 379 F. 2d 1011, 154 U.S.P.Q. 173 (C.C.P.A. 1967).

Even if the references could be combined, they do not teach a level of skill in the art of programmable logic controllers of selecting a part generator, generating the part with the part generator, identifying part locations of the generated part within the manufacturing line, testing the generated part at the part locations, constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability. Applicants are not attacking the references individually, but are clearly pointing out that each reference is deficient and, if combined (although Applicants maintain that they are not combinable), the combination is deficient. The present invention sets forth a unique and non-obvious combination of a method of part flow for a programmable logic controller logical verification system that allows both the controls developer and the information integrator to use the same part flow model present in the VPLC, resulting in substantial cost and time savings.

In addition, the Examiner has adduced no factual basis to support his/her position that it would have been obvious to one of ordinary skill in the art to combine the teachings of Erickson and LeBaron because Erickson describes the need to model the physical system while LeBaron provides teachings related to forming a model of a material delivery system in order to reduce the trial and error involved in developing a model from scratch and benefit from the knowledge available in the prior art. Thus, the Examiner's stated conclusion of obviousness is based on speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis.

The references, if combinable, fail to render obvious the combination of a method of part flow for a programmable logic controller logical verification system including the steps of selecting a part generator, generating the part with the part generator, identifying part locations of the generated part within the manufacturing line, testing the generated part at the part locations, constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability, playing the simulation model by a PLC logical verification system on the computer and viewing a flow of a part through the manufacturing line by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line, determining if the part flow represented in the simulation model is correct to the user, generating PLC code if the part flow represented in the simulation model is correct, using the generated PLC code, and implementing the manufacturing line according to the part flow simulation model as claimed by Applicants. Thus, the Examiner has failed to establish a case of prima facie obviousness.

Against this background, it is submitted that the present invention of claims 2 through 8 is not obvious in view of a proposed combination of Erickson et al. and LeBaron et al. The references cannot be combined to render obvious the combination of the method of part flow for a programmable logic controller logical verification system of claims 2 through 8. Therefore, it is respectfully submitted that claims 2 through 8 are not obvious and are allowable over the rejection under 35 U.S.C. § 103.

### **Claims 13 through 19**

Independent claim 12 claims the present invention as a method for application of a part flow for a programmable logic controller logical verification system. The method includes

the steps of constructing a simulation model of a part flow in a manufacturing line using a computer by representing a part and part locations of the manufacturing line. The method also includes the steps of playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line and viewing a flow of the represented part through the manufacturing line by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line. The method further includes the steps of determining if the part flow represented in the simulation model is correct to the user, generating the PLC code if the part flow simulation model is correct, and using the generated PLC code and implementing the manufacturing line according to the part flow simulation model.

As to the dependent claims, claim 13 claims the method as set forth in claim 12 wherein the step of constructing comprises selecting a part generator. Claim 14 claims the method as set forth in claim 13 wherein the step of constructing further comprises generating a part with the part generator. Claim 15 claims the method as set forth in claim 14 wherein the step of constructing further comprises identifying part locations of the generated part in the manufacturing line. Claim 16 claims the method as set forth in claim 15 wherein the step of constructing further comprises testing the generated part at the part locations. Claim 17 claims the method as set forth in claim 12 wherein the step of constructing comprises constructing a record for the part. Claim 18 claims the method as set forth in claim 17 wherein the record has at least one resource. Claim 19 claims method as set forth in claim 18 wherein the at least one resource has at least one capability.

As to the differences between the prior art and the claims at issue, Erickson et al. merely discloses linking shop-floor hardware such as programmable controllers directly to a

discrete-event simulation model or a graphical animation using a system emulator. Erickson lacks selecting a part generator, generating a part with the part generator, identifying part locations of the generated part in the manufacturing line, testing the generated part at the part locations, and constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability. Erickson et al. also lacks a playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line and viewing a flow of the represented part through the manufacturing line by a user, wherein the PLC logical verification system dynamically interacting through input and output with the simulation model to verify a PLC code of the manufacturing line. In Erickson et al., there is no PLC logical verification system and the simulation model is linked directly with one or more hardware PLCs to test the control logic of the PLC. Also in Erickson et al., one or more hardware PLCs are linked with a system emulator for graphical animation of the manufacturing shop floor. As is known in the art, an emulator represents a physical device in software. (See Wikipedia dictionary definition from Wikipedia website). The PLC logical verification system is not a software representation of a physical device, but a software tool that allows dynamic interaction directly with a simulation model to test PLC logic by having an input and output exchange similar to input/output control logic to validate that the logic is delivering what is intended. The PLC logical verification system analytically verifies the PLC logic. (A similar system is disclosed in U.S. Patent No. 6,442,441). In Erickson et al., the emulator does not allow dynamic interaction directly with a simulation model to test PLC logic to verify a PLC code of the manufacturing line.

LeBaron et al. merely discloses an emulation of a material delivery system in which routing logic, PLC or PC control software, sequencing algorithms and more can be



integrated, tested, and debugged within a simulation environment. LeBaron et al. lacks selecting a part generator, generating a part with the part generator, identifying part locations of the generated part in the manufacturing line, testing the generated part at the part locations, and constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability. LeBaron et al. also lacks playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line and viewing a flow of the represented part through the manufacturing line by a user, wherein the PLC logical verification system dynamically interacting through input and output with the simulation model to verify a PLC code of the manufacturing line. In LeBaron et al., there is no PLC logical verification system and no PLC code is generated. As is known in the art, an emulator represents a physical device in software. (See Wikipedia dictionary definition from Wikipedia website). The PLC logical verification system is not a software representation of a physical device, but a software tool that allows dynamic interaction directly with a simulation model to test PLC logic by having an input and output exchange similar to input/output control logic to validate that the logic is delivering what is intended. The PLC logical verification system analytically verifies the PLC logic. (A similar system is disclosed in U.S. Patent No. 6,442,441). In LeBaron et al., the emulator does not verify the PLC logic or validate that the logic is delivering what is intended.

LeBaron et al. further lacks using the generated PLC code and implementing the manufacturing line according to the part flow simulation model. In LeBaron et al., while LeBaron et al. mentions PLC or PC control software can be tested and debugged within a simulation environment, there is no part flow for a programmable logic controller logical verification system that is viewed by a user and determining if the part flow represented in the

simulation model is correct to the user. Further, in LeBaron et al., there is no generated PLC code that is used in implementing a manufacturing line.

As to a level of ordinary skill in the art, Erickson et al. discloses linking shop-floor hardware such as programmable controllers directly to a discrete-event simulation model or a graphical animation using a system emulator. LeBaron et al. discloses emulation in a delivery system in which routing logic, PLC or PC control software, sequencing algorithms and more can be integrated, tested, and debugged within a simulation environment. There is absolutely no teaching of a level of skill in the programmable logic controller and vehicle manufacturing art that a method of part flow for a programmable logic controller logical verification system includes selecting a part generator, generating a part with the part generator, identifying part locations of the generated part in the manufacturing line, testing the generated part at the part locations, constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability, playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line and viewing a flow of the represented part through the manufacturing line by a user , wherein the PLC logical verification system dynamically interacting through input and output with the simulation model to verify a PLC code of the manufacturing line. The Examiner may not, because he doubts that the invention is patentable, resort to speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis. See In re Warner, 379 F. 2d 1011, 154 U.S.P.Q. 173 (C.C.P.A. 1967).

The present invention sets forth a unique and non-obvious combination of a method of part flow for a programmable logic controller logical verification system that allows both the controls developer and the information integrator to use the same part flow model

present in the VPLC, resulting in substantial cost and time savings. As such, the references, if combinable, fail to render obvious the combination of a method of part flow for a programmable logic controller logical verification system including the steps of viewing a flow of a represented part through the manufacturing line by a user by collecting part locations of the simulation model and coupling the collection of part locations with a part generator as claimed by Applicants in independent claim 12.

In addition, the Examiner has adduced no factual basis to support his/her position that it would have been obvious to one of ordinary skill in the art to combine the teachings of Erickson and LeBaron because Erickson describes the need to model the physical system while LeBaron provides teachings related to forming a model of a material delivery system in order to reduce the trial and error involved in developing a model from scratch and benefit from the knowledge available in the prior art. Thus, the Examiner's stated conclusion of obviousness is based on speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in the factual basis.

The references, if combinable, fail to teach or suggest the combination of a method of part flow for a programmable logic controller logical verification system including the steps of selecting a part generator, generating a part with the part generator, identifying part locations of the generated part in the manufacturing line, testing the generated part at the part locations, and constructing a record for the part wherein the record has at least one resource and wherein the at least one resource has at least one capability, playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line and viewing a flow of the represented part through the manufacturing line by a user , wherein the PLC logical verification system dynamically

interacting through input and output with the simulation model to verify a PLC code of the manufacturing line, determining if the part flow represented in the simulation model is correct to the user, generating PLC code if the part flow represented in the simulation model is correct, using the generated PLC code, and implementing the manufacturing line according to the part flow simulation model as claimed by Applicants.

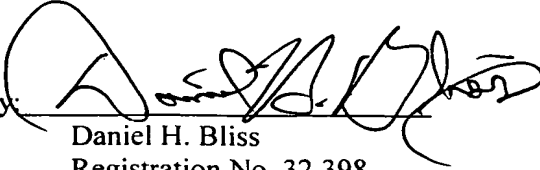
Obviousness under § 103 is a legal conclusion based on factual evidence (In re Fine, 837 F.2d 1071, 1073, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988), and the subjective opinion of the Examiner as to what is or is not obvious, without evidence in support thereof, does not suffice. Since the Examiner has not provided a sufficient factual basis which is supportive of his position (see In re Warner, 379 F.2d 1011, 1017, 154 U.S.P.Q. 173, 178 (C.C.P.A. 1967), cert. Denied, 389 U.S. 1057 (1968)), the rejection of claims 13 through 19 is improper.

Against this background, it is submitted that the present invention of claims 13 through 19 is not obvious in view of a proposed combination of Erickson et al. and LeBaron et al. The references cannot be combined to render obvious the combination of the method of part flow for a programmable logic controller logical verification system of claims 13 through 19. Therefore, it is respectfully submitted that claims 13 through 19 are not obvious and are allowable over the rejection under 35 U.S.C. § 103.

### **CONCLUSION**

In conclusion, it is respectfully submitted that the rejections of claims 1 through 8, 10, and 12 through 21 are improper and should be reversed.

Respectfully submitted,

By:   
Daniel H. Bliss  
Registration No. 32,398

BLISS McGLYNN, P.C.  
2075 West Big Beaver Road  
Suite 600  
Troy, Michigan 48084-3443  
(248) 649-6090

Dated: June 2, 2008

Attorney Docket No.: 0693.00242  
Ford Disclosure No.: 200-0664

**CLAIMS APPENDIX**

The claims on appeal are as follows:

1. A method of part flow for a programmable logic controller logical verification system, said method comprising the steps of:

constructing a simulation model of a manufacturing line using a computer;

playing the simulation model by a PLC logical verification system on the computer and viewing a flow of a part through the manufacturing line by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line;

determining if the part flow represented in the simulation model is correct to the user;

generating the PLC code if the part flow represented in the simulation model is correct; and

using the generated PLC code and implementing the manufacturing line according to the part flow simulation model.

2. A method as set forth in claim 1 wherein said step of constructing comprises selecting a part generator.

3. A method as set forth in claim 2 wherein said step of constructing further comprises generating the part with the part generator.

4. A method as set forth in claim 3 wherein said step of constructing further comprises identifying part locations of the generated part within the manufacturing line.

5. A method as set forth in claim 4 wherein said step of constructing further comprises testing the generated part at the part locations.

6. A method as set forth in claim 1 wherein said step of constructing comprises constructing a record for the part.

7. A method as set forth in claim 6 wherein the record has at least one resource.

8. A method as set forth in claim 7 wherein the at least one resource has at least one capability.

10. A method as set forth in claim 1 including the step of modifying the part flow represented in the simulation model if the part flow represented in the simulation model is not correct.

12. A method for application of a part flow for a programmable logic controller logical verification system, said method comprising the steps of:

constructing a simulation model of a part flow in a manufacturing line using a computer by representing a part and part locations of the manufacturing line;

playing the simulation model by a PLC logical verification system on the computer to move the represented part to and from the part locations within the manufacturing line and viewing a flow of the represented part through the manufacturing line by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line;

determining if the part flow represented in the simulation model is correct to the user;

generating the PLC code if the part flow simulation model is correct; and

using the generated PLC code and implementing the manufacturing line according to the part flow simulation model.

13. A method as set forth in claim 12 wherein said step of constructing comprises selecting a part generator.

14. A method as set forth in claim 13 wherein said step of constructing further comprises generating a part with the part generator.

15. A method as set forth in claim 14 wherein said step of constructing further comprises identifying part locations of the generated part in the manufacturing line.



16. A method as set forth in claim 15 wherein said step of constructing further comprises testing the generated part at the part locations.

17. A method as set forth in claim 12 wherein said step of constructing comprises constructing a record for the part.

18. A method as set forth in claim 17 wherein the record has at least one resource.

19. A method as set forth in claim 18 wherein the at least one resource has at least one capability.

20. A method as set forth in claim 1 including the step of modifying the part flow represented in the simulation model if the part flow represented in the simulation model is not correct.

21. A method for application of a part flow for a programmable logic controller logical verification system, said method comprising the steps of:

constructing a simulation model of a part flow in a manufacturing line using a computer by selecting a part generator, generating a part with the part generator, and identifying part locations of the part in the manufacturing line;

playing the simulation model of the part flow by a PLC logical verification system on the computer to move the generated part to and from locations within the manufacturing line

and viewing a flow of the part through the manufacturing line by a change of color at any of the part locations by a user, wherein the PLC logical verification system dynamically interacts through input and output with the simulation model to verify a PLC code of the manufacturing line;

determining if the part flow represented in the simulation model is correct to the user;

modifying the part flow represented in the simulation model if the part flow represented in the simulation model is not correct;

generating the PLC code if the part flow simulation model is correct; and

using the generated PLC code and implementing the manufacturing line according to the part flow simulation model.

**EVIDENCE APPENDIX**

None

**RELATED PROCEEDINGS APPENDIX**

None